CREATING JAVA PROJECT USING MAVEN

Maven uses **archetype** plugins to create projects. To create a simple java application, we'll use maven-archetype-quickstart plugin. In example below, We'll create a maven based java application project in C:\MVN folder.

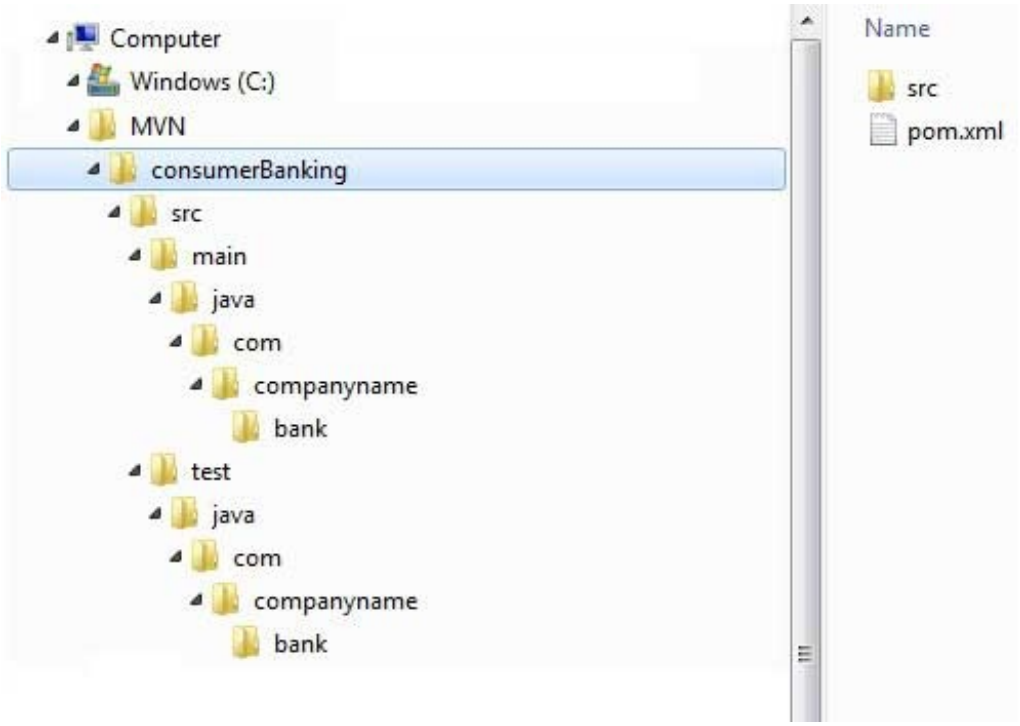
Let's open command console, go the C:\MVN directory and execute the following **mvn** command.

```
C:\MVN>mvn archetype:generate
-DgroupId=com.companyname.bank
-DartifactId=consumerBanking
-DarchetypeArtifactId=maven-archetype-quickstart
-DinteractiveMode=false
```

Maven will start processing and will create the complete java application project structure.

```
INFO] Scanning for projects...
[INFO] Searching repository for plugin with prefix: 'archetype'.
[INFO] ------------------------------------------------------------------------
[INFO] Building Maven Default Project
[INFO]    task-segment: [archetype:generate] (aggregator-style)
[INFO] ------------------------------------------------------------------------
[INFO] Preparing archetype:generate
[INFO] No goals needed for project - skipping
[INFO] [archetype:generate {execution: default-cli}]
[INFO] Generating project in Batch mode
[INFO] ------------------------------------------------------------------------
[INFO] Using following parameters for creating project
 from Old (1.x) Archetype: maven-archetype-quickstart:1.0
[INFO] ------------------------------------------------------------------------
[INFO] Parameter: groupId, Value: com.companyname.bank
[INFO] Parameter: packageName, Value: com.companyname.bank
[INFO] Parameter: package, Value: com.companyname.bank
[INFO] Parameter: artifactId, Value: consumerBanking
[INFO] Parameter: basedir, Value: C:\MVN
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] project created from Old (1.x) Archetype in dir: C:\MVN\consumerBanking
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESSFUL
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 14 seconds
[INFO] Finished at: Tue Jul 10 15:38:58 IST 2012
[INFO] Final Memory: 21M/124M
[INFO] ------------------------------------------------------------------------
```

Now go to C:/MVN directory. You'll see a java application project created named consumerBanking (as specified in artifactId). Maven uses a standard directory layout as shown below:



Using above example, we can understand following key concepts

Folder Structure	Description
consumerBanking	contains src folder and pom.xml
src/main/java	contains java code files under the package structure (com/companyName/bank).
src/main/test	contains test java code files under the package structure (com/companyName/bank).
src/main/resources	it contains images/properties files (In above example, we need to create this structure manually).

If you see, Maven also created a sample Java Source file and Java Test file. Open C:\MVN\consumerBanking\src\main\java\com\companyname\bank folder, you will see App.java.

```
package com.companyname.bank;

/**
 * Hello world!
 *
 */
public class App
{
    public static void main( String[] args )
    {
        System.out.println( "Hello World!" );
    }
}
```

Open C:\MVN\consumerBanking\src\test\java\com\companyname\bank folder, you will see AppTest.java.

```
package com.companyname.bank;

import junit.framework.Test;
import junit.framework.TestCase;
import junit.framework.TestSuite;
```

```
/**
 * Unit test for simple App.
 */
public class AppTest extends TestCase
{
    /**
     * Create the test case
     *
     * @param testName name of the test case
     */
    public AppTest( String testName )
    {
        super( testName );
    }

    /**
     * @return the suite of tests being tested
     */
    public static Test suite()
    {
        return new TestSuite( AppTest.class );
    }

    /**
     * Rigourous Test :-)
     */
    public void testApp()
    {
        assertTrue( true );
    }
}
```

Developers are required to place their files as mentioned in table above and Maven handles the all the build related complexities.

In next section, we'll discuss how to build and test the project using maven Maven Build & Test Project.